



IFW
PATENT

Atty. Docket No. 8729-227 (IB-200306-022)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT(S): Tae Gon Park
SERIAL NO.: 10/721,398
FILED: November 25, 2003
FOR: CRYPTOGRAPHIC SYSTEMS AND METHODS
SUPPORTING MULTIPLE MODES


Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF PRIORITY DOCUMENT

Sir:

Enclosed is a certified copy of Korean Appln. No. 2003/0004806
filed on January 24, 2003 and Korean Appln. No. 2003/0053262 filed on July 31, 2003
and from which priority is claimed under 35 U.S.C. §119.

Respectfully submitted,



Frank V. DeRosa
Reg. No. 43,584
Attorney for Applicant(s)

F. CHAU & ASSOCIATES, LLC
130 Woodbury Road
Woodbury, NY 11797
(516) 692-8888

대한민국 특허청
KOREAN INTELLECTUAL
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 : 10-2003-0004806
Application Number

출원년월일 : 2003년 01월 24일
Date of Application JAN 24, 2003

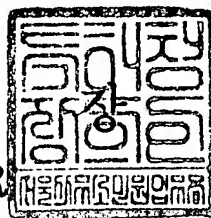
출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 12 월 11 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0004
【제출일자】	2003.01.24
【발명의 명칭】	다수의 동작 모드들을 지원하는 암호화 장치
【발명의 영문명칭】	CRYPTOGRAPHIC APPARATUS FOR SUPPORTING MULTIPLE MODES
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	임창현
【대리인코드】	9-1998-000386-5
【포괄위임등록번호】	1999-007368-2
【대리인】	
【성명】	권혁수
【대리인코드】	9-1999-000370-4
【포괄위임등록번호】	1999-056971-6
【발명자】	
【성명의 국문표기】	박태건
【성명의 영문표기】	PARK, TAE GON
【주민등록번호】	700915-1068320
【우편번호】	442-706
【주소】	경기도 수원시 팔달구 망포동 동수원엘지빌리지 108동 106호
【국적】	KR
【발명자】	
【성명의 국문표기】	남경완
【성명의 영문표기】	NAM, KYUNG WAN
【주민등록번호】	720411-1347511
【우편번호】	463-060
【주소】	경기도 성남시 분당구 이매동 한신아파트 209동 1407호
【국적】	KR

【발명자】

【성명의 국문표기】

박영욱

【성명의 영문표기】

PARK, YOUNG WOOK

【주민등록번호】

730625-1055314

【우편번호】

430-012

【주소】

경기도 안양시 만안구 안양2동 838-14 7/5

【국적】

KR

【심사청구】

청구

【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인

임창현 (인) 대리인

권혁수 (인)

【수수료】

【기본출원료】

20 면 29,000 원

【가산출원료】

30 면 30,000 원

【우선권주장료】

0 건 0 원

【심사청구료】

21 항 781,000 원

【합계】

840,000 원

【첨부서류】

1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 메모리에 저장된 데이터를 암호화하기 위한 암호화 장치에 관한 것으로, 작고 간단한 소자들을 이용하여 ECB, CBC, CBC-MAC, 카운터 및 OCB 모드 등에서 동작할 수 있는 암호화 장치가 개시된다. 또한, 본 발명의 암호화 장치는, CPU와 암호화 장치 간의 데이터 송수신을 최소화함으로써 통신 시스템의 성능을 향상시킨다. 한편, 암호화 장치 내에 구성되는 입력 버퍼 및 출력 버퍼가 각각 적어도 2 개의 블록들을 저장할 수 있도록 구성함으로써 암호화 장치의 성능을 극대화할 수 있다. 더욱이, 암호화 장치가 제로-패딩 기능을 지원함으로써 CPU가 처리해야할 작업을 최소화한다.

【대표도】

도 2

【명세서】

【발명의 명칭】

다수의 동작 모드들을 지원하는 암호화 장치{CRYPTOGRAPHIC APPARATUS FOR SUPPORTING MULTIPLE MODES}

【도면의 간단한 설명】

도 1은 본 발명의 바람직한 실시예에 따른 암호화 장치를 구비한 통신 시스템을 보여주는 도면;

도 2는 도 1에 도시된 암호화 장치의 상세한 구성을 보여주는 블록도;

도 3 내지 도 5는 도 2에 도시된 블록 암호화 유닛의 동작 모드에 따른 회로 구성을 보여주는 도면들;

도 4는 도 2에 도시된 블록 암호화 유닛이 CBC 모드 또는 CBC-MAC 모드로 동작하기 위해 필요한 회로 구성들을 보여주는 도면;

도 5는 도 2에 도시된 블록 암호화 유닛이 CNT 모드로 동작하기 위해 필요한 회로 구성들을 보여주는 도면;

도 6은 본 발명의 바람직한 실시예에 따른 블록 암호화 유닛을 보여주는 도면;

도 7은 도 1에 도시된 암호화 장치와 메모리 사이의 데이터 송수신을 예시적으로 보여주는 도면;

도 8은 도 2에 도시된 DMA 컨트롤러의 FSM(Finite State Machine)을 보여주는 도면;

도 9는 도 2에 도시된 입력 버퍼, 출력 버퍼 그리고 블록 암호화 유닛의 관계에 따른 FSM을 보여주는 도면;

도 10a 내지 도 10b는 데이터의 마지막에 '0'을 삽입하는 제로-패딩(zero-padding)을 예시적으로 보여주는 도면;

도 11은 OCB 모드의 암호화 프로세스를 보여주는 도면;

12는 OCB 모드의 복호화 프로세스를 개념적으로 보여주는 도면; 그리고

도 13은 도 1에 도시된 메모리의 소스 어드레스들(SA0 - SAm+1)에 저장된 데이터 블록들(I0 - Im+1)과 목표 어드레스들(DA0 - DAm+1)에 저장된 데이터 블록들(O0 - Om+1)을 보여주는 도면이다.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<14> 본 발명은 데이터 암호화에 관한 것으로, 좀 더 구체적으로는 데이터를 암호화/복호화하는 암호화 장치에 관한 것이다.

<15> 암호화 기술은 메시지 전송의 안전을 보장하기 위하여 자주 사용되고 있다. 암호화 기술은 전송측(transmission side)에서 메시지(이하, 플레인텍스트(plaintext))를 부호화(encrypt)하고, 수신측(receiving side)에서 메시지(이하, 싸이퍼텍스트(ciphering))를 해독(decrypt) 또는 복호화(decode)한다. 이러한 메시지의 부호화 및 복호화는 암호화 기술로 널리 알려져 있다.

<16> 데이터 암호화 표준(Data Encryption Standard : DES)은 여러 나라들과

ANSI(American National Standards Institute)에서 표준으로 채용된 블록단위 암호화 프로토콜이다. 이외에도 암호화 프로토콜에는 3-DES 및 AES 등이 있다. 블록단위 암호화 프로토콜에는 여러가지 동작 모드들 즉, ECB(Electronic Codebook), CBC(Cipher Block Chaining), OFB(Output Feedback), 및 CFB(Cipher Feedback) 등을 정의하고 있다. 최근에는 카운터(counter) 모드 및 OCB(Offset Codebook) 모드 등도 제안되었다.

<17> 이와 같이 다양한 동작 모드들은 개별적인 하드웨어로 구현될 수 있다. 그러나, 다양한 동작 모드들을 하나의 칩으로 구현하기 위해서는 많은 수의 게이트들이 필요하다.

【발명이 이루고자 하는 기술적 과제】

<18> 따라서 본 발명의 목적은 다수의 동작 모드들을 지원하는 암호화 장치를 제공하는데 있다.

<19> 본 발명의 다른 목적은 간단한 회로 구성을 가지면서 다수의 동작 모드들을 지원하는 암호화 장치를 제공하는데 있다.

<20> 본 발명의 또다른 목적은 다수의 동작 모드들을 지원하는 통신 시스템을 제공하는데 있다.

【발명의 구성 및 작용】

<21> (구성)

<22> 상술한 바와 같은 목적을 달성하기 위한 본 발명의 특징에 의하면, 메모리에 저장된 데이터를 암호화하기 위한 암호화 장치는: 외부로부터 제공되는 암호화 정보에 응답해서 제어 신호들을 생성하는 제어 유닛과, 상기 메모리로부터 상기 데이터를 독출하는 메모리 컨트롤러와, 상기 메모리 컨트롤러에 의해서 독출된 데이터를 저장하기 위한 입력 버퍼와, 상기 입력 버퍼

에 저장된 데이터 블록을 암호화하기 위한 암호화 유닛 그리고 상기 암호화 유닛에 의해 암호화된 데이터를 저장하기 위한 출력 버퍼를 포함한다. 상기 메모리 컨트롤러는 상기 출력 버퍼에 저장된 상기 암호화된 데이터를 상기 메모리에 기입하고, 상기 메모리 컨트롤러, 상기 입력 버퍼, 상기 암호화 유닛 및 상기 출력 버퍼는 상기 제어 신호들에 응답해서 동작한다.

<23> 바람직한 실시예에 있어서, 상기 제어 유닛은, 상기 암호화 정보에 응답해서 초기 벡터 및 동작 모드를 나타내는 동작 모드 신호를 생성한다.

<24> 이 실시예에 있어서, 상기 동작 모드는, ECB(Electronic Codebook) 모드, CBC(Cipher Block Chaining) 모드, CBC-MAC(CBC-Message Authentication Code) 모드 또는 CNT(Counter) 모드 중 어느 하나이다.

<25> 이 실시예에 있어서, 상기 암호화 유닛은, 상기 입력 버퍼로부터 제공된 데이터를 저장하기 위한 데이터 입력 레지스터와, 상기 데이터 입력 레지스터에 저장된 데이터를 암호화하는 암호화기 그리고 상기 암호화기에 의해서 암호화된 데이터를 저장하기 위한 데이터 출력 레지스터를 포함한다.

<26> 이 실시예에 있어서, 상기 암호화 유닛은 상기 ECB 모드로 동작한다.

<27> 바람직한 실시예에 있어서, 상기 암호화 유닛은, 상기 제어 유닛으로부터 제공된 초기 벡터를 저장하기 위한 초기 벡터 레지스터와, 상기 초기 벡터 레지스터에 저장된 상기 초기 벡터와 상기 데이터 입력 레지스터에 저장된 데이터를 논리 연산하는 제 1 논리 연산기를 더 포함한다.

<28> 이 실시예에 있어서, 상기 암호화 유닛은 상기 CBC 또는 CBC-MAC 모드 중 어느 하나로 동작한다.



- <29> 이 실시예에 있어서, 상기 암호화 유닛이 상기 CBC-MAC 모드로 동작할 때, 상기 메모리 컨트롤러는 상기 출력 버퍼에 저장된 암호화된 데이터를 상기 메모리에 기입하지 않는다.
- <30> 바람직한 실시예에 있어서, 상기 암호화 유닛은, 상기 제어 유닛으로부터 제공된 초기 벡터를 저장하기 위한 초기 벡터 레지스터와, 상기 초기 벡터 레지스터에 저장된 데이터에 미리 설정된 값을 더해서 상기 초기 벡터 레지스터에 다시 저장하는 가산기와, 상기 초기 벡터 레지스터에 저장된 데이터를 암호화하는 암호화기와, 상기 입력 버퍼로부터 제공된 데이터를 저장하기 위한 데이터 입력 레지스터와, 상기 암호화기에 의해서 암호화된 데이터와 상기 데이터 입력 레지스터에 저장된 데이터를 논리 연산하기 위한 제 2 논리 연산기 그리고 상기 제 2 논리 연산기의 출력을 저장하기 위한 데이터 출력 레지스터를 포함한다.
- <31> 이 실시예에 있어서, 상기 암호화 유닛은 상기 CNT 모드로 동작한다.
- <32> 바람직한 실시예에 있어서, 상기 암호화 유닛은, 상기 제어 유닛으로부터의 초기 벡터를 저장하는 초기 벡터 레지스터와, 상기 입력 버퍼로부터의 상기 데이터를 저장하는 데이터 입력 레지스터와, 입력 데이터를 암호화하는 암호화기와, 상기 출력 버퍼로 제공될 데이터를 저장하는 데이터 출력 레지스터와, 가산기와, 상기 동작 모드 신호에 응답해서, 상기 초기 벡터 레지스터에 저장된 데이터, 상기 초기 벡터 레지스터에 저장된 데이터 및 상기 데이터 입력 레지스터에 저장된 데이터를 논리 연산한 결과 데이터, 또는 상기 데이터 입력 레지스터에 저장된 데이터 가운데 하나를 상기 암호화기의 입력 데이터로서 제공하고, 상기 동작 모드 신호에 응답해서, 상기 초기 벡터 레지스터에 저장된 데이터를 상기 가산기로 전달하는 제 1 선택 회로 그리고 상기 동작 모드 신호에 응답해서, 상기 데이터 입력 레지스터에 저장된 데이터 및 상기 블록 암호화기에 의해서 암호화된 데이터를 논리 연산한 결과 데이터 또는 상기 블록 암호화기에 의해서 암호화된 데이터를 상기 데이터 출력 레지스터로 전달하고, 상기 동작 모드 신호에

응답해서, 상기 암호화기에 의해서 암호화된 데이터를 상기 초기 벡터 레지스터로 전달하는 제 2 선택 회로를 포함한다. 상기 가산기는 상기 제 1 선택 회로로부터 제공된 데이터에 미리 설정된 값을 더해서 상기 초기 벡터 레지스터에 저장한다.

<33> 이 실시예에 있어서, 상기 제 1 선택 회로는, 제 1 멀티플렉서와, 제 1 논리 연산기와, 상기 모드 신호에 응답해서, 상기 초기 벡터 레지스터에 저장된 데이터를 상기 가산기와 상기 제 1 멀티플렉서로 또는 상기 제 1 논리 연산기로 전달하는 제 2 멀티플렉서 그리고 상기 모드 신호에 응답해서, 상기 데이터 입력 레지스터에 저장된 데이터를 상기 제 1 논리 연산기 또는 상기 제 1 멀티플렉서로 전달하는 제 3 멀티플렉서를 포함한다. 상기 제 1 논리 연산기는 상기 제 2 및 제 3 멀티플렉서들로부터의 출력들을 받아들여 논리 연산하고, 상기 제 1 멀티플렉서는 상기 모드 신호에 응답해서, 상기 제 2 멀티플렉서, 상기 제 1 논리 연산기 또는 상기 제 3 멀티플렉서의 출력을 상기 블록 암호화기로 전달한다. 상기 제 2 선택 회로는, 제 4 멀티플렉서, 제 2 논리 연산기, 그리고 상기 모드 신호에 응답해서, 상기 암호화기의 출력을 상기 제 2 논리 연산기 또는 상기 제 4 멀티플렉서 또는/그리고 상기 초기 벡터 레지스터로 전달하는 제 5 멀티플렉서를 포함한다. 상기 제 2 논리 연산기는 상기 제 3 및 제 5 멀티플렉서들의 출력들을 받아들여서 논리 연산하고, 상기 제 4 멀티플렉서는 상기 제 2 논리 연산기 또는 상기 제 5 멀티플렉서의 출력을 상기 데이터 출력 데이터로 전달한다.

<34> 바람직한 실시예에 있어서, 상기 제어 유닛은, 상기 입력 버퍼에 저장된 데이터는 블록 단위로 상기 암호화 유닛으로 제공되도록 제어한다.

<35> 이 실시예에 있어서, 상기 제어 유닛은, 상기 입력 버퍼 저장된 마지막 데이터가 미리 설정된 블록 크기보다 작을 때 제로-패딩(zero-padding)을 수행한다.

- <36> 바람직한 실시예에 있어서, 상기 입력 버퍼는 FIFO(First-In First-Out) 버퍼이고, 적어도 2 개의 데이터 블록들을 저장한다.
- <37> 바람직한 실시예에 있어서, 상기 출력 버퍼는 FIFO(First-In First-Out) 버퍼이고, 적어도 2 개의 암호화된 데이터 블록들을 저장한다.
- <38> 바람직한 실시예에 있어서, 상기 메모리 컨트롤러는, DMA(Direct Memory Access) 컨트롤러이다.
- <39> 본 발명의 다른 특징에 의하면, 통신 시스템은: 메모리와, 상기 메모리에 저장된 데이터를 암호화하기 위한 암호화 장치 그리고 상기 암호화 장치의 동작을 제어하는 중앙 처리 장치를 포함한다. 상기 암호화 장치는, 상기 중앙 처리 장치로부터 제공되는 암호화 정보에 응답해서 제어 신호들을 생성하는 제어 유닛과, 상기 메모리로부터 상기 데이터를 독출하는 메모리 컨트롤러와, 상기 메모리 컨트롤러에 의해서 독출된 데이터를 저장하기 위한 입력 버퍼와, 상기 입력 버퍼에 저장된 데이터 블록을 암호화하기 위한 암호화 유닛 그리고 상기 암호화 유닛에 의해 암호화된 데이터를 저장하기 위한 출력 버퍼를 포함한다. 상기 메모리 컨트롤러는 상기 출력 버퍼에 저장된 상기 암호화된 데이터를 상기 메모리에 기입하고, 상기 메모리 컨트롤러, 상기 입력 버퍼, 상기 암호화 유닛 및 상기 출력 버퍼는 상기 제어 신호들에 응답해서 동작한다.
- <40> (실시예)
- <41> 이하 본 발명의 바람직한 실시예를 첨부된 도면들을 참조하여 상세히 설명한다.
- <42> 도 1은 본 발명의 바람직한 실시예에 따른 암호화 장치를 구비한 통신 시스템을 보여주는 도면이다. 암호화 장치를 구비한 통신 시스템에는 데스크 탑 컴퓨터, 휴대용 컴퓨터, PDA

및 핸드폰 등이 있다. 도 1을 참조하면, 통신 시스템은, CPU(10), 메모리(20), 암호화 장치(30), 중재기(40) 그리고 시스템 버스(50)를 포함한다. 메모리(20)는 암호화 프로그램을 저장하고 있으며, 플레인텍스트와 싸이퍼텍스트를 저장한다. CPU(10)는 메모리(20)에 저장된 암호화 프로그램에 따라서 암호화 장치(30)가 동작하도록 제어한다. 중재기(40)는 시스템 버스(50)에 여러 개의 DMA 마스터들이 연결되어 있을 때, 버스(50)를 사용할 수 있는 권한을 특정 DMA 마스터에게 부여한다. 도 1에 도시된 통신 시스템에서는 CPU(10)와 암호화 유닛(30)이 DMA 마스터로서 기능하기 때문에 중재기(40)가 필요하다.

<43> 도 2는 도 1에 도시된 암호화 장치(30)의 상세한 구성을 보여주는 블록도이다. 도 2를 참조하면, 암호화 장치(30)는 제어 유닛(31), DMA 컨트롤러(32), 입력 버퍼(33), 출력 버퍼(34) 그리고 블록 암호화 유닛(block cipher unit)(35)을 포함한다.

<44> 제어 유닛(31)은 CPU(10)로부터 제공되는 암호화 정보 즉, 소스 어드레스(Source Address : SA), 목적 어드레스(Destination Address : DA), 데이터 크기(data size : D_SIZE), 블록 크기(block size : B_SIZE), 키(key : K), 키 사이즈(key size), 방향(direction)(암호화(encryption) 또는 복호화(decryption)), 초기 벡터(Initialization Vector) 및 동작 모드가 입력되면, 암호화 장치(30) 내의 구성 요소들을 제어한다. DMA 컨트롤러(32)는 제어 유닛(31)의 제어에 응답해서, 메모리(20)의 소스 어드레스(SA)에 저장된 플레인텍스트를 독출해서 입력 버퍼(33)에 저장하고 그리고 출력 버퍼(34)에 저장된 싸이퍼텍스트를 메모리(20)의 목표 어드레스(DA)에 저장한다. 입력 버퍼(33)에 저장된 플레인텍스트는 블록 단위로 블록 암호화 유닛(35)으로 전달된다. 예컨대, 블록 암호화 유닛(35)이 128 비트 AES로 구성되는 경우, 입력 버퍼(33)로부터 블록 암호화 유닛(35)으로 전달되는 블록의 크기는 128 비트이다. 출력 버퍼(34)는 블록 암호화 유닛(35)으로부터의 싸이퍼텍스트 블록을 저장

하고, 제어 유닛(31)의 제어에 따라서 저장된 싸이퍼텍스트를 DMA 컨트롤러(32)로 전달한다. 입력 버퍼(33) 및 출력 버퍼(34)는 각각 FIFO(First-In First-Out) 버퍼로 구성되고, 크기는 블록 암호화 유닛(35)에서 처리하는 블록의 크기의 2배이다. 그러므로, 입력 버퍼(33) 및 출력 버퍼(34)는 2 개의 블록들을 각각 저장할 수 있다. 예컨대, 블록 암호화 유닛(35)이 한 번에 처리하는 플레인텍스트 블록의 크기가 128 비트(bit)이면, 입력 버퍼(33) 및 출력 버퍼(34)의 크기는 각각 256 비트이다. 이 실시예에서, 입력 버퍼(33) 및 출력 버퍼(34)는 각각 2 개의 블록들을 저장할 수 있는 것으로 설명하였으나, 입력 버퍼(33) 및 출력 버퍼(34)의 크기는 다양하게 변경될 수 있다. 블록 암호화 유닛(35)은 예컨대, AES(Advanced Encryption Standard) 블록 암호화기를 포함한다. 블록 암호화 유닛(35)의 상세한 회로 구성 및 동작은 이하 상세히 설명된다.

<45> 도 3 내지 도 5는 도 2에 도시된 블록 암호화 유닛(35)의 동작 모드에 따른 회로 구성을 보여주는 도면들이다. 먼저, 도 3은 도 2에 도시된 블록 암호화 유닛(35)이 ECB(Electronic Codebook) 모드로 동작하기 위해 필요한 회로 구성들을 보여주고 있다. 도 3을 참조하면, ECB 모드의 블록 암호화 유닛(35a)은 데이터 입력 레지스터(110), 데이터 출력 레지스터(120) 그리고 블록 암호화기(130)를 포함한다. 도 2에 도시된 입력 버퍼(33)로부터의 플레인텍스트 블록은 데이터 입력 레지스터(110)를 통하여 블록 암호화기(130)로 입력된다. 블록 암호화기(130)는 제어 유닛(31)으로부터 주어지는 키(K)에 따라서 암호화를 수행하고, 싸이퍼텍스트를 데이터 출력 레지스터(120)로 출력한다. 데이터 출력 레지스터(120)에 저장된 데이터는 도 2에 도시된 출력 버퍼(34)로 출력된다.

- <46> 도 4는 도 2에 도시된 블록 암호화 유닛(35)이 CBC(Cipher Block Chaining) 모드 또는 CBC-MAC(CBC-Message Authentication Code) 모드로 동작하기 위해 필요한 회로 구성들을 보여주고 있다.
- <47> 도 4를 참조하면, 블록 암호화 유닛(35b)은 초기값 레지스터(210), 데이터 입력 레지스터(220), 데이터 출력 레지스터(230), 익스클루시브-오아 연산기(240) 그리고 블록 암호화기(250)를 포함한다. 데이터 입력 레지스터(220)를 통하여 도 2에 도시된 입력 버퍼(33)로부터의 플레인텍스트 블록과 초기값 레지스터(210)에 설정된 초기 벡터(initialization vector) 데이터는 익스클루시브-오아 연산기(240)에서 익스클루시브-오아 연산된다. 초기값 레지스터(210)에 설정되는 초기 벡터 데이터는 제어 유닛(31)으로부터 제공된다. 익스클루시브-오아 연산기(240)의 연산 결과는 블록 암호화기(250)로 전달된다. 블록 암호화기(250)는 제어 유닛(31)으로부터 제공된 키(K)에 따라서 익스클루시브-오아 연산기(240)의 연산 결과를 암호화한다. 블록 암호화기(250)로부터의 싸이퍼텍스트는 데이터 출력 레지스터(230)와 초기값 레지스터(210)에 저장된다. 데이터 출력 레지스터(230)에 저장된 데이터는, CBC 모드일 때 도 2의 출력 버퍼(34)로 출력되나, CBC-MAC 모드일 때 출력 버퍼(34)로 출력되지 않는다. CBC-MAC 모드에서는, 메모리(20)에 저장된 플레인텍스트들이 모두 암호화되고 나서 초기값 레지스터(210)에 저장된 최종 데이터만이 출력 버퍼(34)로 출력된다.
- <48> 도 5는 도 2에 도시된 블록 암호화 유닛(35)이 CNT(Counter) 모드로 동작하기 위해 필요한 회로 구성들을 보여주고 있다. 도 5를 참조하면, 블록 암호화 유닛(35c)은, 레지스터(310), 데이터 입력 레지스터(320), 데이터 출력 레지스터(330), 가산기(340), 블록 암호화기(250) 그리고 익스클루시브-오아 연산기(350)를 포함한다.

<49> 레지스터(310)에는 제어 유닛(31)으로부터 제공된 초기 데이터가 저장된다. 가산기(340)는 레지스터(310)에 저장된 데이터에 1을 더한다. 가산기(340)의 출력은 다시 레지스터(310)에 저장된다. 블록 암호화기(350)는 제어 유닛(31)으로부터 제공된 키(K)에 따라서 레지스터(310)에 저장된 데이터를 암호화한다. 데이터 입력 레지스터(320)는 도 2에 도시된 입력 버퍼(33)로부터의 플레인텍스트 블록을 저장한다. 익스클루시브-오아 연산기(350)는 블록 암호화기(350)로부터의 출력과 데이터 입력 레지스터(320)에 저장된 플레인텍스트를 익스클루시브-오아 연산하고 그 결과를 데이터 출력 레지스터(330)에 저장한다. 데이터 출력 레지스터(330)에 저장된 데이터는 싸이퍼텍스트로서, 도 2의 출력 버퍼(34)로 출력된다.

<50> 도 3 내지 도 5에 도시된 바와 같이, 블록 암호화 유닛(35)은 암호화 모드들에 따라서 조금씩 다른 회로 구성을 필요로 한다. 그러나, 각각의 모드를 위한 회로들을 개별적으로 구성한다면 회로 면적이 커질 것이다. 본 발명에서는 간단한 회로 구성을 가지면서도 앞서 설명한 암호화 모드들에서 모두 동작될 수 있는 블록 암호화 유닛을 제공한다. 도 6은 본 발명의 바람직한 실시예에 따른 블록 암호화 유닛(35)을 보여주고 있다.

<51> 도 6을 참조하면, 블록 암호화 유닛(35)은 레지스터(410), 데이터 입력 레지스터(420), 데이터 출력 레지스터(430), 가산기(440), 멀티플렉서들(450-454), 익스클루시브-오아 연산기들(461, 462) 그리고 블록 암호화기(470)를 포함한다. WLAN에 본 발명이 적용되는 경우, 도 6에 도시된 블록 암호화기(470)는 AES(Advanced Encryption Standard) 블록 암호화기로 구성되거나 다른 응용 분야에서는 DES(Data Encryption Standard) 또는 3-DES 등 다른 블록 암호화기로 구성될 수 있다.

<52> 레지스터(410)는 CBC, CBC-MAC 및 CNT 모드에서 사용되며, 도 2의 제어 유닛(31)에 의해 초기값이 설정된다. 데이터 입력 레지스터(420)는 도 2의 입력 버퍼(33)로부터 입력되는 플레

인텍스트 블록을 저장한다. 데이터 출력 레지스터(430)는 멀티플렉서(452)로부터 출력되는 싸이퍼텍스트를 저장한다. 멀티플렉서들(450-454)은 제어 유닛(31)으로부터의 모드 신호(MD)에 응답해서 각각 동작한다. 모드 신호(MD)는 블록 암호화 유닛(35)의 동작 모드를 나타내는 신호로서, 다수의 비트들로 구성된다. 이 실시예에서, 블록 암호화 유닛(35)은 CBC, CBC-MAC 또는 CNT 모드로 동작할 수 있으므로, 모드 신호(MD)는 2비트이다. 예컨대, 모드 신호(MD)가 '00'이면 ECB 모드, '01'이면 CBC 모드, '10'이면 CBC-MAC 모드 그리고 '11'이면 CNT 모드를 나타낸다. 도 6에서, 굵은 점선은 CNT 모드에서 데이터 이동 경로를 나타내고, 굵은 실선은 CBC 또는 CBC-MAC 모드에서 데이터 이동 경로 그리고 가는 실선은 ECB 모드에서 데이터 이동 경로를 나타낸다. 그리고 가는 실선은 모든 모드들에서 데이터 이동 경로를 나타낸다.

<53> 멀티플렉서(450)는 모드 신호(MD)가 CNT 모드를 나타낼 때 레지스터(410)에 저장된 데이터를 가산기(440)와 멀티플렉서(453)로 전달하고, 모드 신호(MD)가 CBC 모드 또는 CBC-MAC 모드를 나타낼 때 레지스터(410)에 저장된 데이터를 익스클루시브-오아 연산기(461)로 전달한다. 멀티플렉서(451)는 모드 신호(MD)가 CBC 모드 또는 CBC-MAC 모드를 나타낼 때, 레지스터(420)에 저장된 플레인텍스트 블록을 익스클루시브-오아 연산기(461)로 전달하고, 모드 신호(MD)가 ECB 모드를 나타낼 때 레지스터(420)에 저장된 플레인텍스트 블록을 멀티플렉서(453)로 전달하고, 그리고 모드 신호(MD)가 CNT 모드를 나타낼 때 레지스터(420)에 저장된 플레인텍스트 블록을 익스클루시브-오아 연산기(461)로 전달한다. 멀티플렉서(453)는 모드 신호(MD)에 응답해서 멀티플렉서들(450, 512) 또는 익스클루시브-오아 연산기(461)의 출력 가운데 하나를 블록 암호화기(470)로 전달한다. 즉, 멀티플렉서(453)는, 모드 신호(MD)가 CNT 모드를 나타낼 때 멀티플렉서(450)의 출력을, 모드 신호(MD)가 CBC 또는 CBC-MAC 모드를 나타낼 때 익스클루시브-

오아 연산기(461)의 출력을, 그리고 모드 신호(MD)가 ECB 모드를 나타낼 때 멀티플렉서(451)의 출력을 블록 암호화기(470)로 전달한다.

<54> 멀티플렉서(454)는 모드 신호(MD)에 응답해서 블록 암호화기(470)로부터 출력되는 싸이퍼텍스트를 익스클루시브-오아 연산기(462), 멀티플렉서(452) 또는/그리고 레지스터(410)로 전달한다. 즉, 멀티플렉서(454)는, 모드 신호(MD)가 CNT 모드를 나타낼 때 블록 암호화기(470)로부터의 싸이퍼텍스트를 익스클루시브-오아 연산기(462)로, 모드 신호(MD)가 ECB 모드를 나타낼 때 블록 암호화기(470)로부터의 싸이퍼텍스트를 멀티플렉서(452)로 그리고 모드 신호(MD)가 CBC 및 CBC-MAC 모드를 나타낼 때 블록 암호화기(470)로부터의 싸이퍼텍스트를 멀티플렉서(452) 및 레지스터(410)로 전달한다. 멀티플렉서(452)는 모드 신호(MD)가 CNT 모드를 나타낼 때 익스클루시브-오아 연산기(462)로부터의 연산 결과를 그리고 ECB, CBC 또는 CBC-MAC 모드일 때 멀티플렉서(454)의 출력을 데이터 출력 레지스터(430)로 전달한다.

<55> 상술한 바와 같이 본 발명의 블록 암호화 유닛(35)은, 레지스터들(410, 420, 430), 가산기(440), 멀티플렉서들(451-454), 익스클루시브-오아 연산기들(461, 462) 그리고 블록 암호화기(470)를 포함하여 ECB, CBC, CBC-MAC 및 CNT 모드를 모두 수행할 수 있다.

<56> 도 7은 도 1에 도시된 암호화 장치(30)와 메모리(20) 사이의 데이터 송수신을 예시적으로 보여주고 있다. 암호화 장치(30)는 CPU(10)로부터 제공된 소스 어드레스(SA), 목표 어드레스(DA) 및 데이터 크기(D_SIZE) 등에 따라서 메모리(20)를 액세스한다. 도 7에 도시된 바와 같이, 암호화 장치(30)는 메모리(20)의 소스 어드레스(SA)에 저장된 플레인텍스트(plaintext)를 독출해서 암호화를 수행한다. 그리고, 암호화 장치(30)는 암호화된 싸이퍼텍스트(ciphertext)를 메모리(20)의 목표 어드레스(DA)에 저장한다. 암호화 장치(30)는 제 0 독출 단계(R0)에서 독출된 플레인텍스트(PT0)에 대한 암호화가 완료되면 제 0 기입 단계(W0)에서 싸



이퍼텍스트(CT0)를 메모리(20)에 기입한다. 이와 같은 방법으로, 나머지 독출 단계들(R1, R2 및 R3)과 나머지 기입 단계들(W0, W1 및 W2)이 순차적으로 수행된다. 그러나, 암호화 장치(30)가 제 0 독출 단계(R0)를 수행한 후에 제 0 기입 단계(W0)를 수행하고, 다시 제 1 독출 단계(R1)를 수행하는 것은 암호화 장치(30)의 성능 저하를 유발한다. 왜냐하면, 암호화 장치(30)가 메모리(20)에 싸이퍼텍스트를 기입하는 동안 암호화 장치(30) 내의 블록 암호화 유닛(35)은 어떠한 동작도 안하고 쉬게 되기 때문이다. 본 발명에서는 이러한 문제를 해결하기 위하여 앞서 설명한 바와 같이, 입력 버퍼(33)와 출력 버퍼(34)(도 2 참조)를 각각 블록 사이즈의 2 배 크기로 구성한다. 그러므로, R0, W0, R1 및 W1과 같이 독출 단계와 기입 단계를 순차적으로 하지 않고, 독출 단계와 기입 단계를 다양하게 변형할 수 있다. 독출 및 기입 방법에 관해서 이하 상세히 설명한다.

<57> 앞서 설명한 바와 같이, 암호화 장치(30)가 최대의 성능을 발휘할 수 있도록 블록 암호화 유닛(35)의 쉬는 시간(idle time)을 최소화해야만 한다. 그러기 위해서는 입력 데이터의 결핍(starvation) 상태를 방지하는 것이 중요하다. 본 발명에서는 입력 버퍼(33)의 크기가 블록 암호화 유닛(35)으로 입력되는 블록의 크기의 2 배 즉, 입력 버퍼(33)가 2 개의 블록들을 저장한다. 그러므로, 블록 암호화 유닛(35)이 암호화 프로세스를 수행하는 동안 입력 버퍼(33)에는 새로운 데이터가 기입될 수 있다. 또한, 블록 암호화 유닛(35)으로부터 출력된 싸이퍼텍스트가 출력 버퍼(34)에 임시 저장됨으로서 메모리(20)에 싸이퍼텍스트가 기입되기 전이라도 블록 암호화 유닛(35)은 다음 플레인텍스트 블록에 대한 암호화 프로세스를 수행할 수 있게 되는 것이다.

<58> 한편, DMA 컨트롤러(32)는 메모리(20)로부터 플레인텍스트를 독출하는 동작과 싸이퍼텍스트를 메모리(20)에 기입하는 동작을 동시에 수행할 수 없다. 그러므로, 제어 유닛(31)은 미

리 설정된 우선 순위에 따라서 DMA 컨트롤러(32)가 독출 및 기입 프로세스를 수행하도록 제어한다. 다음 표 1은 입력 버퍼(33)와 출력 버퍼(34)에 저장된 블록의 개수에 따른 DMA 컨트롤러(32)의 동작 우선 순위를 보여주고 있다.

<59> 【표 1】

입력 버퍼(33)에 저장된 블록 수 (IBUFCNT)	출력 버퍼(34)에 저장된 블록 수 (OBUFCNT)	DMA 프로세스
2	0	NOP
1	0	독출
0	0	독출
2	1	기입
1	1	독출
0	1	독출
2	2	기입
1	2	기입
0	2	독출

<60> 표 1의 기본 개념은 입력 버퍼(33)에 빈 공간이 있으면 독출 동작에 우선 순위를 주고, 출력 버퍼(34)에 적어도 하나의 블록이 저장되어 있으면 기입 동작에 우선 순위를 주는 것이다. 표 1에서, 입력 버퍼(33)가 완전히 비어 있고, 출력 버퍼(34)가 완전히 채워져 있을 때 블록 암호화 유닛(35)이 동작하지 않고 있다면, 입력 버퍼(33)에 플레인텍스트 블록을 넣는 독출 동작을 수행하여 블록 암호화 유닛(35)이 동작할 수 있도록 제어한 후, 출력 버퍼(34)에 저장된 싸이퍼텍스트를 메모리(20)에 기입하는 것이 바람직하다. 그러나, 입력 버퍼(33)가 완전히 비어 있고, 출력 버퍼(34)가 완전히 채워져 있을 때 블록 암호화 유닛(35)이 동작하고 있다면 독출 또는 기입 동작 중 어느 것을 먼저 수행하더라도 무방하다. 그러나, 동

작의 기준은 필요하므로, 이러한 경우에는 DMA 컨트롤러(32)가 독출 동작을 수행하는 것으로 설정한다. 표 1의 NOP(no operation)는 DMA 컨트롤러(32)가 어떠한 동작도 수행하지 않음을 나타낸다.

- <61> 이와 같이, 입력 버퍼(33)와 출력 버퍼(34)가 각각 적어도 2 개의 블록들을 저장할 수 있도록 구성함으로써 시스템 버스(50)가 동작(BUSY) 상태에 있더라도 암호화 유닛(35)은 버퍼들(33, 34)을 이용하여 블록 암호화 프로세스를 수행할 수 있다.
- <62> 도 8은 도 2에 도시된 DMA 컨트롤러의 FSM(Finite State Machine)을 보여준다. 도 8에서, 독출 제어 신호(MORE_DATAR)는, 메모리(20)의 소스 어드레스(SA)로부터 독출해서 암호화 프로세스를 수행해야 할 전체 데이터의 크기(D_SIZE)가 현재까지 DMA 컨트롤러(32)에 의해서 메모리(20)로부터 독출된 데이터의 크기보다 클 때 1, 그리고 작거나 같을 때 0이다. 기입 제어 신호(MORE_DATAW)는, 암호화 프로세스가 수행된 후 메모리(20)의 목표 어드레스(DA)에 기입될 전체 데이터의 크기(DATA_SIZE)가 현재까지 DMA 컨트롤러에 의해서 기입된 데이터의 크기보다 클 때 1, 그리고 작거나 같을 때 0이다.
- <63> 제어 유닛(31)은 다음 수학적 식 1을 만족할 때 DMA 컨트롤러(32)가 휴지 상태(510)에서 독출 상태(520)로 천이하도록 제어한다.
- <64> 【수학적 식 1】 $\text{MORE_DATAR} \ \&\& \ (\text{RBUFCNT} \neq 2) \ \&\& \ (\text{RBUFCNT} \neq 1 \ \parallel \ \text{WBUFCNT} \neq 2) = 1$
- <65> DMA 컨트롤러(32)는 메모리(20)로부터 데이터를 독출하는 동작(DMA_READ)이 완료되면 독출 완료 신호(DMA_READ_DONE)를 활성화하고, 독출 상태(520)에서 휴지 상태(510)로 천이한다. 이 때, 독출 카운트(DMA_READ_CNT) 값이 1만큼 증가한다.

<66> 제어 유닛(31)은 다음 수학적 식 2를 만족할 때 DMA 컨트롤러(32)가 휴지 상태(510)에서 독출 상태(530)로 천이한다.

<67> **【수학적 식 2】** $\text{MORE_DATAW} \ \&\& \ ((\text{RBUFCNT}=2) \parallel (\text{RBUFCNT}=1 \ \&\& \ \text{WBUFCNT}=2))$
 $\&\& \ \text{!CBC-MAC} = 1$

<68> DMA 컨트롤러(32)는 메모리(20)에 데이터를 기입하는 동작(DMA_WRITE)이 완료되면 기입 완료 신호(DMA_WRITE_DONE)를 활성화하고, 기입 상태(530)에서 휴지 상태(510)로 천이한다. 이 때, 기입 카운트(DMA_WRITE_CNT) 값이 1만큼 증가한다. 독출 완료 신호(DMA_READ_DONE)와 기입 완료 신호(DMA_WRITE_DONE)는 제어 유닛(31)으로 제공되며, 독출 카운트(DMA_READ_CNT)와 기입 카운트(DMA_WRITE_CNT)는 제어 유닛(31) 내에 구성된 카운터들(미 도시됨)의 값이다.

<69> 메모리(20)로부터 독출될 데이터 크기(DATA_SIZE)와 독출 카운트(DMA_READ_CNT)의 관계에 따른 독출 제어 신호(MORE_DATAR)의 상태 그리고 메모리(20)에 기입될 데이터 크기(DATA_SIZE)와 기입 카운트(DMA_WRITE_CNT)의 관계에 따른 기입 제어 신호(MORE_DATAW)의 상태가 다음 표 2 및 표 3에 각각 정리되어 있다.

<70> **【표 2】**

독출 제어 신호 (MORE_DATAR)	조 건
0	DATA_SIZE < DMA_READ_CNT
1	DATA_SIZE > DMA_READ_CNT

<71>

【표 3】

기입 제어 신호 (MORE_DATAW)	조 건
0	DATA_SIZE < DMA_WRITE_CNT
1	DATA_SIZE > DMA_WRITE_CNT

<72> 도 9는 도 2에 도시된 입력 버퍼(33), 출력 버퍼(34) 그리고 블록 암호화 유닛(35)의 관
계에 따른 FSM을 보여주고 있다. 도 9를 참조하면, 제어 유닛(31)은, 블록 암호화 유닛(35)에
서 처리해야할 데이터가 남아있으면(즉, 독출 제어 신호(MORE_DATAR)가 1이면) 블록 암호화 유
닛(35)을 휴지 상태(610)에서 블록 암호화 유닛(35) 체크 상태(620)로 천이시킨다. 블록 암호
화 유닛(35)은 암호화 프로세스를 시작할 준비가 되었을 때 입력 준비 신호(INPUT_READY)를 1
로 활성화하고, 현재 입력된 블록에 대한 암호화 프로세스가 종료되었을 때 종료 신호
(OUTPUT_READY)를 1로 활성화한다.

<73> 체크 상태(620)에 놓인 블록 암호화 유닛(350)은, 입력 준비 신호(INPUT_READY)가 1이고
그리고 입력 버퍼(33)에 적어도 하나의 데이터가 있으면, 입력 버퍼(33)로부터 블록 암호화 유
닛(35)으로 데이터를 전송하는 상태(640)로 천이한다. 입력 버퍼(33)로부터 블록 암호화
유닛(35)으로 데이터 전송이 완료되면 블록 암호화 유닛(35)은 입력 완료 신호(INPUT_DONE)를
활성화하고, 다시 체크 상태(620)로 천이한다.

<74> 체크 상태(620)에 놓인 블록 암호화 유닛(350)은, 출력 준비 신호(OUTPUT_READY)가 1이
고 그리고 출력 버퍼(33)가 꽉 차 있지 않으면, 블록 암호화 유닛(35)으로부터 출력 버퍼(34)
로 데이터를 전송하는 상태(640)로 천이한다. 블록 암호화 유닛(35)으로부터 출력 버퍼(34)로

데이터 전송이 완료되면 블록 암호화 유닛(35)은 출력 완료 신호(OUTPUT_DONE)을 활성화하고, 다시 체크 상태(620)로 천이한다.

<75> 도 8 및 도 9에서, CBC_MAC 모드인 경우, 각 블록에 대한 암호 프로세스가 완료되었을 때 생성되는 싸이퍼텍스트는 불필요하기 때문에 메모리(20)에 기입할 필요가 없다. 그러므로, 암호화 장치(30)의 시스템 버스(50) 사용을 최소화하기 위하여 CBC-MAC 모드에서는 각 블록에 대한 싸이퍼텍스트를 메모리(20)에 기입하지 않는다. CBC-MAC 모드에서는 모든 블록들에 대한 암호화 프로세스가 종료되었을 때 레지스터(410)에 저장된 데이터를 메모리(20)의 목표 어드레스(DA)에 한 번 기입하기만 하면 된다.

<76> 도 10a 내지 도 10b는 데이터의 마지막에 '0'을 삽입하는 제로-패딩(zero-padding)을 예시적으로 보여주고 있다. 도 10a를 참조하면, 데이터 프레임이 n 개의 블록들로 구성되고, 1 블록의 크기는 L1일 때 마지막 n 번째 블록의 크기는 항상 L1이 아니다. 예컨대, n 번째 블록의 크기 L2가 L1보다 작으면 n 번째 블록의 크기가 L1이 되도록 데이터의 마지막에 '0'을 삽입하는 제로-패딩이 수행된다. 본 발명에서는 CPU(10)와 암호화 장치(20) 사이의 데이터 송수신을 최소화하기 위하여 암호화 장치(20) 내에서 제로-패딩을 수행한다. 즉, 제어 유닛(31)은 초기에 CPU(10)로부터 수신되는 데이터 사이즈가 블록 사이의 정수배가 아니면 DMA 컨트롤러(32)를 통해 입력 버퍼(33)로 입력된 마지막 블록에 제로-패딩을 수행한다. 도 10b에 도시된 바와 같이, 제어 유닛(31)은 입력 버퍼(33)에 저장된 n 번째 블록의 마지막에 '0'들을 삽입하여 n 번째 블록의 크기가 L1이 되도록 한다.

<77> 정리하면, CPU(10)와 암호화 장치(20) 사이에 송수신되는 데이터는 다음과 같다. 우선, CPU(10)는 제어 정보를 암호화 장치(30)의 제어 유닛(31)으로 전달한다. 제어 정보에는 소스

어드레스, 목표 어드레스, 데이터 사이즈, 키, 키 사이즈, 방향(암호화 또는 복호화), 초기 벡터 그리고 동작 모드가 있다.

<78> 기본적으로, 암호화될 데이터 즉, 플레인텍스트는 메모리(20)의 소스 어드레스에 저장되어 있어야 한다. 그리고 나서 CPU(10)가 제어 정보를 암호화 장치(30)에게 알려주면 암호화 장치(30)는 암호화 프로세스를 수행하고, 처리 결과인 싸이퍼텍스트를 목표 어드레스에 저장한다. 이와 같은 본 발명에 의하면, CPU(10)와 암호화 장치(30) 사이의 상호 접속이 최소화되어서 시스템의 성능 저하를 방지한다.

<79> 한편, 동작 모드에 따라서 CPU(10)와 암호화 장치(30) 사이에 송수신되는 제어 정보와 데이터가 다르다. ECB 모드에서, 플레인텍스트는 메모리(20)의 소스 어드레스에 저장되어 있어야 하며, 싸이퍼텍스트는 메모리(20)의 목표 어드레스에 저장된다. ECB 모드에서 소스 어드레스와 목표 어드레스는 동일해도 무방하다. CPU(10)가 암호화 장치(30)로 제공해야 할 제어 정보는 소스 어드레스, 목표 어드레스, 키, 키 사이즈, 데이터 사이즈, 방향 그리고 동작 모드이다. ECB 모드에서는, 각 블록에 대한 암호화 프로세스가 완료될 때마다 싸이퍼텍스트가 메모리(20)의 목표 어드레스에 저장되어야 한다.

<80> CBC 모드는 ECB 모드와 유사하나, 제어 정보에 초기 벡터가 포함되어야 한다. CBC-MAC 모드에서는 각 블록에 대한 싸이퍼텍스트는 불필요하기 때문에 매 블록마다 싸이퍼텍스트를 메모리(20)에 기입하지 않아도 된다. 다만, 모든 블록들에 대한 암호화 프로세스가 종료되고 나서 레지스터(410)에 저장된 데이터가 CPU(10)로 전송되어야 한다.

<81> OCB(offset codebook) 모드는 CPU(10)의 연산을 많이 필요로 한다. 도 11은 OCB 모드의 암호화 프로세스를 그리고 도 12는 OCB 모드의 복호화 프로세스를 개념적으로 보여주고 있다. OCB 모드에 관한 상세한 내용은 미국공개공보 2000-51537에 개시되어 있다. 본 발명의 암호화

장치(30)는 미리 작성된 코드북(codebook)과 오프셋(offset)에 따라서 각 블록의 싸이퍼텍스트를 생성한다. 도 11에서, 인출 번호 710 및 720으로 묶여진 부분은 CPU(10)에 의해 수행되고, 인출 번호 730, 740 및 750으로 묶여진 제 1 내지 제 3 스테이지들은 암호화 장치(30)에 의해 수행된다.

<82> 도 13은 도 1에 도시된 메모리(20)의 소스 어드레스들(SA0 - SAm+1)에 저장된 데이터 블록들(I0 - Im+1)과 목표 어드레스들(DA0 - DAm+1)에 저장된 데이터 블록들(O0 - Om+1)을 보여주고 있다.

<83> 도 11 및 도 13을 참조하면, 암호화 장치(30)는 OCB 모드동안 ECB 모드와 동일하게 동작한다. 제 1 스테이지(730)에서, 메모리(20)의 소스 어드레스(I0)에 저장된 데이터 블록(I0)은 데이터 입력 레지스터(420)(도 6)에 저장된다. 블록 암호화기(470)는 데이터 입력 레지스터(420)에 저장된 데이터 블록(I0)에 대한 암호화 프로세스를 수행하고 그 결과를 데이터 출력 레지스터(450)에 저장한다. 데이터 출력 레지스터(450)에 저장된 데이터는 오프셋(Offset0)으로서 CPU(10)로 전달된다. CPU(10)는 암호화 장치(30)로부터 제공된 오프셋(Offset0)과 미리 설정된 코드북(Lntz(1) - Lntz(m))에 따라서 오프셋들(Offset1 - Offsetm)을 계산하고, 메모리(20)의 소정 영역에 저장된 플레인텍스트들(M1-Mm)과 계산된 오프셋들(Offset1 - Offsetm)을 익스클루시브-오아 연산하고, 연산 결과를 메모리(20)의 소스 어드레스들(SA1 - SAm+1)에 저장한다.

<84> 제 2 스테이지(740)에서, 암호화 장치(30)는 메모리(20)의 소스 어드레스들(SA1 - SAm+1)에 저장된 데이터 블록들을 순차적으로 독출해서 암호화 프로세스를 수행하고, 암호화된 데이터 블록들(O0 - Om+1)을 메모리(20)의 목표 어드레스들(DA1 - DAm+1)에 저장한다.

- <85> CPU(10)는 암호화 장치(30)의 제 2 스테이지(740)가 완료되면 목표 어드레스들(DA1 - DAm+1)에 저장된 데이터 블록들(00 - 0m+1) 그리고 오프셋들(Offset1 - Offsetm-1) 및 플레인 텍스트(Mm)에 대한 익스클루시브-오아 연산을 수행하여 싸이퍼텍스트들(C1 - Cm)을 생성한다. 그리고, CPU(10)는 체크섬(Checksum)과 오프셋(Offsetm)에 대한 익스클루시브-오아 연산을 수행해서 소스 어드레스(DAm+1)에 저장한다.
- <86> 제 3 스테이지에서, 암호화 장치(30)는 소스 어드레스(SAm+1)에 저장된 데이터 블록 (Im+1)을 독출해서 암호화 프로세스를 수행하고, 암호화된 데이터 블록(0m+1)을 메모리(20)의 목표 어드레스(DAm+1)에 저장한다. CPU(10)는 메모리(20)의 목표 어드레스(DAm+1)에 저장된 데이터(0m+1)를 독출하고, 데이터(0m+1)의 일부를 MIC로서 취한다.
- <87> 도 12를 참조하여, OCB 모드의 복호화 과정이 설명된다. 도 12에서, 인출 번호 810 및 820으로 묶여진 부분은 CPU(10)에 의해 수행되고, 인출 번호 830, 840, 850 및 860으로 묶여진 제 1 내지 제 4 스테이지들은 암호화 장치(30)에 의해 수행된다. OCB 모드의 복호화 과정은 도 11에 도시된 OCB 모드의 암호화 과정의 역순으로 진행되나, 암호화 장치(30)의 4 개의 스테이지들(830-060)을 통해서 복호화가 수행된다. 여기서, OCB 모드의 복호화 역시 암호화 장치(30)를 ECB 모드의 복호화로 설정한 후 수행된다.
- <88> 예시적인 바람직한 실시예를 이용하여 본 발명을 설명하였지만, 본 발명의 범위는 개시된 실시예들에 한정되지 않는다는 것이 잘 이해될 것이다. 오히려, 본 발명의 범위에는 다양한 변형 예들 및 그 유사한 구성들이 모두 포함될 수 있도록 하려는 것이다. 따라서, 청구범위는 그러한 변형 예들 및 그 유사한 구성들 모두를 포함하는 것으로 가능한 폭넓게 해석되어야 한다.

【발명의 효과】

<89> 이와 같은 본 발명에 의하면, 작고 간단한 소자들을 이용하여 ECB, CBC, CBC-MAC, 카운터 및 OCB 모드 등에서 동작할 수 있는 암호화 장치가 구현된다. 또한, CPU와 암호화 장치 간의 데이터 송수신을 최소화함으로써 통신 시스템의 성능을 향상시킨다. 한편, 암호화 장치 내에 구성되는 입력 버퍼 및 출력 버퍼가 각각 적어도 2 개의 블록들을 저장할 수 있도록 구성함으로써 암호화 장치의 성능을 극대화할 수 있다. 더욱이, 암호화 장치가 제로-패딩 기능을 지원함으로써 CPU가 처리해야할 작업을 최소화한다.

【특허청구범위】**【청구항 1】**

메모리에 저장된 데이터를 암호화하기 위한 암호화 장치에 있어서:

외부로부터 제공되는 암호화 정보에 응답해서 제어 신호들을 생성하는 제어 유닛과;

상기 메모리로부터 상기 데이터를 독출하는 메모리 컨트롤러와;

상기 메모리 컨트롤러에 의해서 독출된 데이터를 저장하기 위한 입력 버퍼와;

상기 입력 버퍼에 저장된 데이터 블록을 암호화하기 위한 암호화 유닛; 그리고

상기 암호화 유닛에 의해 암호화된 데이터를 저장하기 위한 출력 버퍼를 포함하되;

상기 메모리 컨트롤러는 상기 출력 버퍼에 저장된 상기 암호화된 데이터를 상기 메모리에 기입하고;

상기 메모리 컨트롤러, 상기 입력 버퍼, 상기 암호화 유닛 및 상기 출력 버퍼는 상기 제어 신호들에 응답해서 동작하는 것을 특징으로 하는 암호화 장치.

【청구항 2】

제 1 항에 있어서,

상기 제어 유닛은,

상기 암호화 정보에 응답해서 초기 벡터 및 동작 모드를 나타내는 동작 모드 신호를 생성하는 것을 특징으로 하는 암호화 장치.

【청구항 3】

제 2 항에 있어서,

상기 동작 모드는, ECB(Electronic Codebook) 모드, CBC(Cipher Block Chaining) 모드, CBC-MAC(CBC-Message Authentication Code) 모드 또는 CNT(Counter) 모드 중 어느 하나인 것을 특징으로 하는 암호화 장치.

【청구항 4】

제 3 항에 있어서,

상기 암호화 유닛은,

상기 입력 버퍼로부터 제공된 데이터를 저장하기 위한 데이터 입력 레지스터와;

상기 데이터 입력 레지스터에 저장된 데이터를 암호화하는 암호화기; 그리고

상기 암호화기에 의해서 암호화된 데이터를 저장하기 위한 데이터 출력 레지스터를 포함하는 것을 특징으로 하는 암호화 장치.

【청구항 5】

제 4 항에 있어서,

상기 암호화 유닛은 상기 ECB 모드로 동작하는 것을 특징으로 하는 암호화 장치.

【청구항 6】

제 4 항에 있어서,

상기 암호화 유닛은,

상기 제어 유닛으로부터 제공된 초기 벡터를 저장하기 위한 초기 벡터 레지스터와;

상기 초기 벡터 레지스터에 저장된 상기 초기 벡터와 상기 데이터 입력 레지스터에 저장된 데이터를 논리 연산하는 제 1 논리 연산기를 더 포함하는 것을 특징으로 하는 암호화 장치.

【청구항 7】

제 6 항에 있어서,

상기 암호화 유닛은 상기 CBC 또는 CBC-MAC 모드 중 어느 하나로 동작하는 것을 특징으로 하는 암호화 장치.

【청구항 8】

제 7 항에 있어서,

상기 암호화 유닛이 상기 CBC-MAC 모드로 동작할 때,

상기 메모리 컨트롤러는 상기 출력 버퍼에 저장된 암호화된 데이터를 상기 메모리에 기입하지 않는 것을 특징으로 하는 암호화 장치.

【청구항 9】

제 3 항에 있어서,

상기 암호화 유닛은,

상기 제어 유닛으로부터 제공된 초기 벡터를 저장하기 위한 초기 벡터 레지스터와;

상기 초기 벡터 레지스터에 저장된 데이터에 미리 설정된 값을 더해서 상기 초기 벡터 레지스터에 다시 저장하는 가산기와;

상기 초기 벡터 레지스터에 저장된 데이터를 암호화하는 암호화기와;

상기 입력 버퍼로부터 제공된 데이터를 저장하기 위한 데이터 입력 레지스터와;

상기 암호화기에 의해서 암호화된 데이터와 상기 데이터 입력 레지스터에 저장된 데이터를 논리 연산하기 위한 제 2 논리 연산기; 그리고

상기 제 2 논리 연산기의 출력을 저장하기 위한 데이터 출력 레지스터를 포함하는 것을 특징으로 하는 암호화 장치.

【청구항 10】

제 9 항에 있어서,

상기 암호화 유닛은 상기 CNT 모드로 동작하는 것을 특징으로 하는 암호화 장치.

【청구항 11】

제 3 항에 있어서,

상기 암호화 유닛은,

상기 제어 유닛으로부터의 초기 벡터를 저장하는 초기 벡터 레지스터와;

상기 입력 버퍼로부터의 상기 데이터를 저장하는 데이터 입력 레지스터와;

입력 데이터를 암호화하는 암호화기와;

상기 출력 버퍼로 제공될 데이터를 저장하는 데이터 출력 레지스터와;

가산기와;

상기 동작 모드 신호에 응답해서, 상기 초기 벡터 레지스터에 저장된 데이터, 상기 초기 벡터 레지스터에 저장된 데이터 및 상기 데이터 입력 레지스터에 저장된 데이터를 논리 연산한 결과 데이터, 또는 상기 데이터 입력 레지스터에 저장된 데이터 가운데 하나를 상기 암호화기의 입력 데이터로서 제공하고, 상기 동작 모드 신호에 응답해서, 상기 초기 벡터 레지스터에 저장된 데이터를 상기 가산기로 전달하는 제 1 선택 회로; 그리고

상기 동작 모드 신호에 응답해서, 상기 데이터 입력 레지스터에 저장된 데이터 및 상기 블록 암호화기에 의해서 암호화된 데이터를 논리 연산한 결과 데이터 또는 상기 블록 암호화기

에 의해서 암호화된 데이터를 상기 데이터 출력 레지스터로 전달하고, 상기 동작 모드 신호에 응답해서, 상기 암호화기에 의해서 암호화된 데이터를 상기 초기 벡터 레지스터로 전달하는 제 2 선택 회로를 포함하되;

상기 가산기는 상기 제 1 선택 회로로부터 제공된 데이터에 미리 설정된 값을 더해서 상기 초기 벡터 레지스터에 저장하는 것을 특징으로 하는 암호화 장치.

【청구항 12】

제 11 항에 있어서,

상기 제 1 선택 회로는,

제 1 멀티플렉서와;

제 1 논리 연산기와;

상기 모드 신호에 응답해서, 상기 초기 벡터 레지스터에 저장된 데이터를 상기 가산기와 상기 제 1 멀티플렉서로 또는 상기 제 1 논리 연산기로 전달하는 제 2 멀티플렉서; 그리고

상기 모드 신호에 응답해서, 상기 데이터 입력 레지스터에 저장된 데이터를 상기 제 1 논리 연산기 또는 상기 제 1 멀티플렉서로 전달하는 제 3 멀티플렉서를 포함하되;

상기 제 1 논리 연산기는 상기 제 2 및 제 3 멀티플렉서들로부터의 출력들을 받아들여 논리 연산하고;

상기 제 1 멀티플렉서는 상기 모드 신호에 응답해서, 상기 제 2 멀티플렉서, 상기 제 1 논리 연산기 또는 상기 제 3 멀티플렉서의 출력을 상기 블록 암호화기로 전달하는 것을 특징으로 하는 암호화 장치.

【청구항 13】

제 12 항에 있어서,

상기 제 2 선택 회로는,

제 4 멀티플렉서와;

제 2 논리 연산기; 그리고

상기 모드 신호에 응답해서, 상기 암호화기의 출력을 상기 제 2 논리 연산기 또는 상기 제 4 멀티플렉서 또는/그리고 상기 초기 벡터 레지스터로 전달하는 제 5 멀티플렉서를 포함하
되;

상기 제 2 논리 연산기는 상기 제 3 및 제 5 멀티플렉서들의 출력들을 받아들여서 논리
연산하고;

상기 제 4 멀티플렉서는 상기 제 2 논리 연산기 또는 상기 제 5 멀티플렉서의 출력을 상
기 데이터 출력 데이터로 전달하는 것을 특징으로 하는 암호화 장치.

【청구항 14】

제 1 항에 있어서,

상기 제어 유닛은,

상기 입력 버퍼에 저장된 데이터는 블록 단위로 상기 암호화 유닛으로 제공되도록 제어
하는 것을 특징으로 하는 암호화 장치.

【청구항 15】

제 14 항에 있어서,

상기 제어 유닛은,

상기 입력 버퍼 저장된 마지막 데이터가 미리 설정된 블록 크기보다 작을 때 제로-패딩 (zero-padding)을 수행하는 것을 특징으로 하는 암호화 장치.

【청구항 16】

제 1 항에 있어서,

상기 입력 버퍼는 FIFO(First-In First-Out) 버퍼인 것을 특징으로 하는 암호화 장치.

【청구항 17】

제 1 항에 있어서,

상기 입력 버퍼는 적어도 2 개의 데이터 블록들을 저장하는 것을 특징으로 하는 암호화 장치.

【청구항 18】

제 1 항에 있어서,

상기 출력 버퍼는 FIFO(First-In First-Out) 버퍼인 것을 특징으로 하는 암호화 장치.

【청구항 19】

제 1 항에 있어서,

상기 출력 버퍼는 적어도 2 개의 암호화된 데이터 블록들을 저장하는 것을 특징으로 하는 암호화 장치.

【청구항 20】

제 1 항에 있어서,

상기 메모리 컨트롤러는,

DMA(Direct Memory Access) 컨트롤러인 것을 특징으로 하는 암호화 장치.

【청구항 21】

통신 시스템에 있어서:

메모리와 ;

상기 메모리에 저장된 데이터를 암호화하기 위한 암호화 장치; 그리고

상기 암호화 장치의 동작을 제어하는 중앙 처리 장치를 포함하되;

상기 암호화 장치는,

상기 중앙 처리 장치로부터 제공되는 암호화 정보에 응답해서 제어 신호들을 생성하는 제어 유닛과;

상기 메모리로부터 상기 데이터를 독출하는 메모리 컨트롤러와;

상기 메모리 컨트롤러에 의해서 독출된 데이터를 저장하기 위한 입력 버퍼와;

상기 입력 버퍼에 저장된 데이터 블록을 암호화하기 위한 암호화 유닛; 그리고

상기 암호화 유닛에 의해 암호화된 데이터를 저장하기 위한 출력 버퍼를 포함하되;

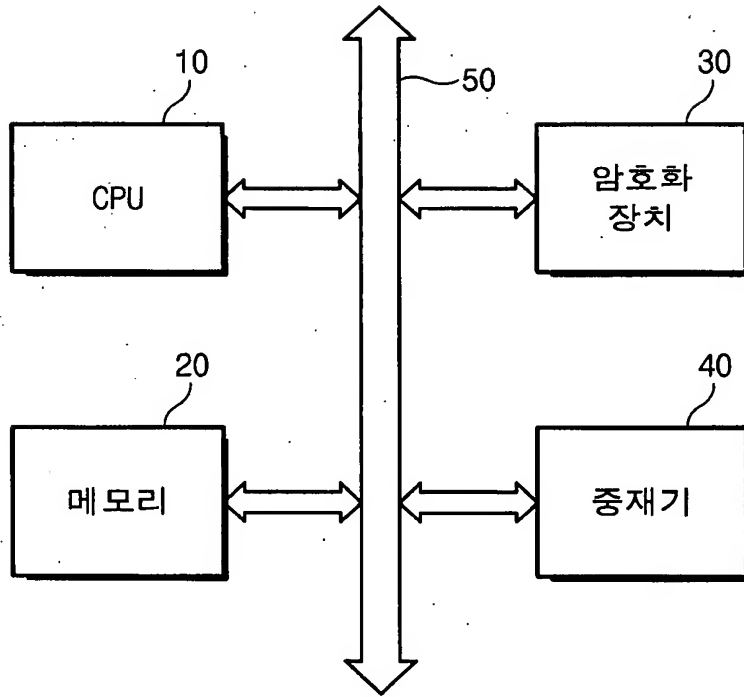
상기 메모리 컨트롤러는 상기 출력 버퍼에 저장된 상기 암호화된 데이터를 상기 메모리에 기입하고;

상기 메모리 컨트롤러, 상기 입력 버퍼, 상기 암호화 유닛 및 상기 출력 버퍼는 상기 제어 신호들에 응답해서 동작하는 것을 특징으로 하는 암호화 장치.

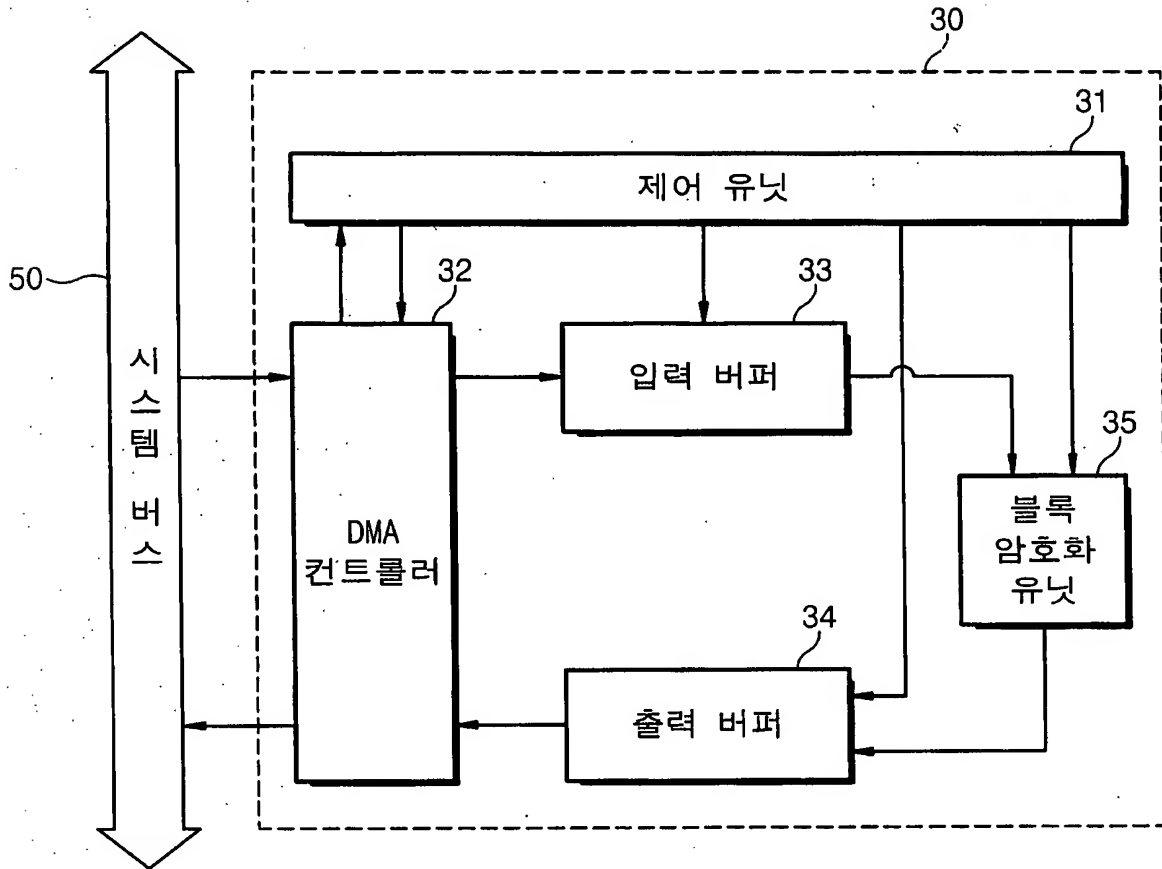


【도면】

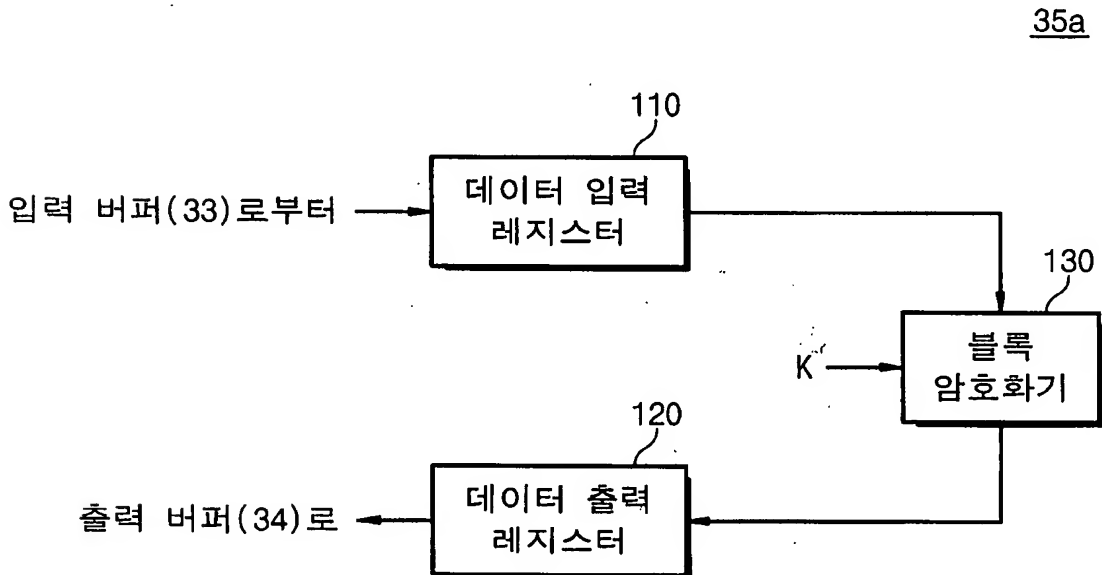
【도 1】



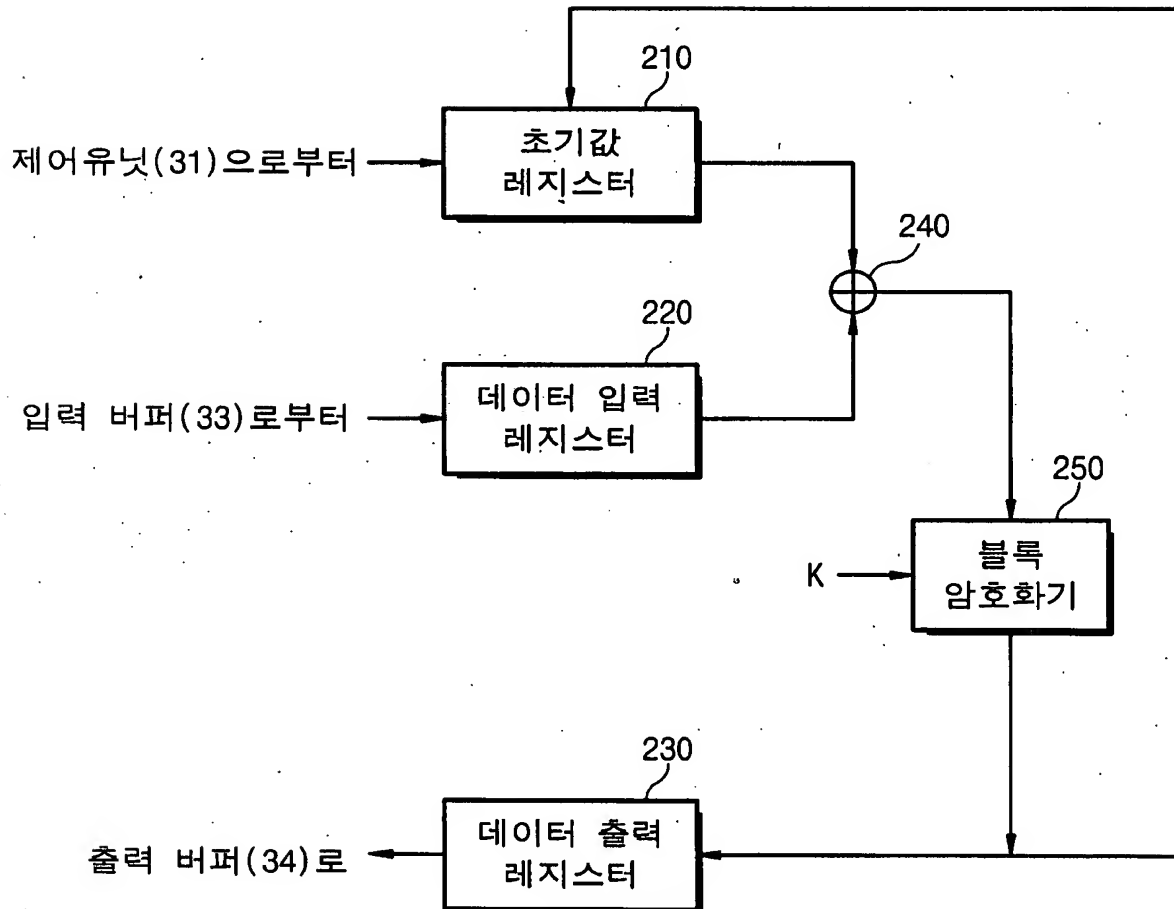
【도 2】



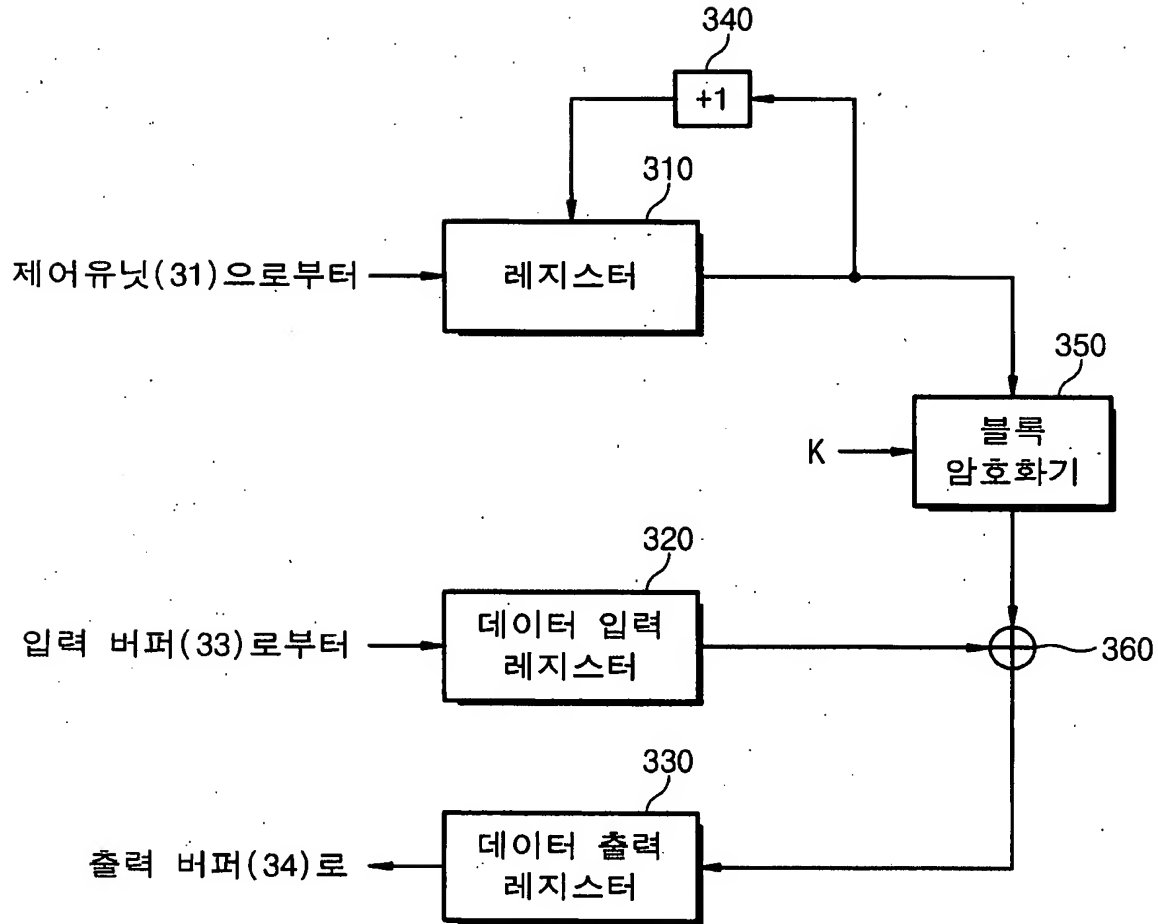
【도 3】



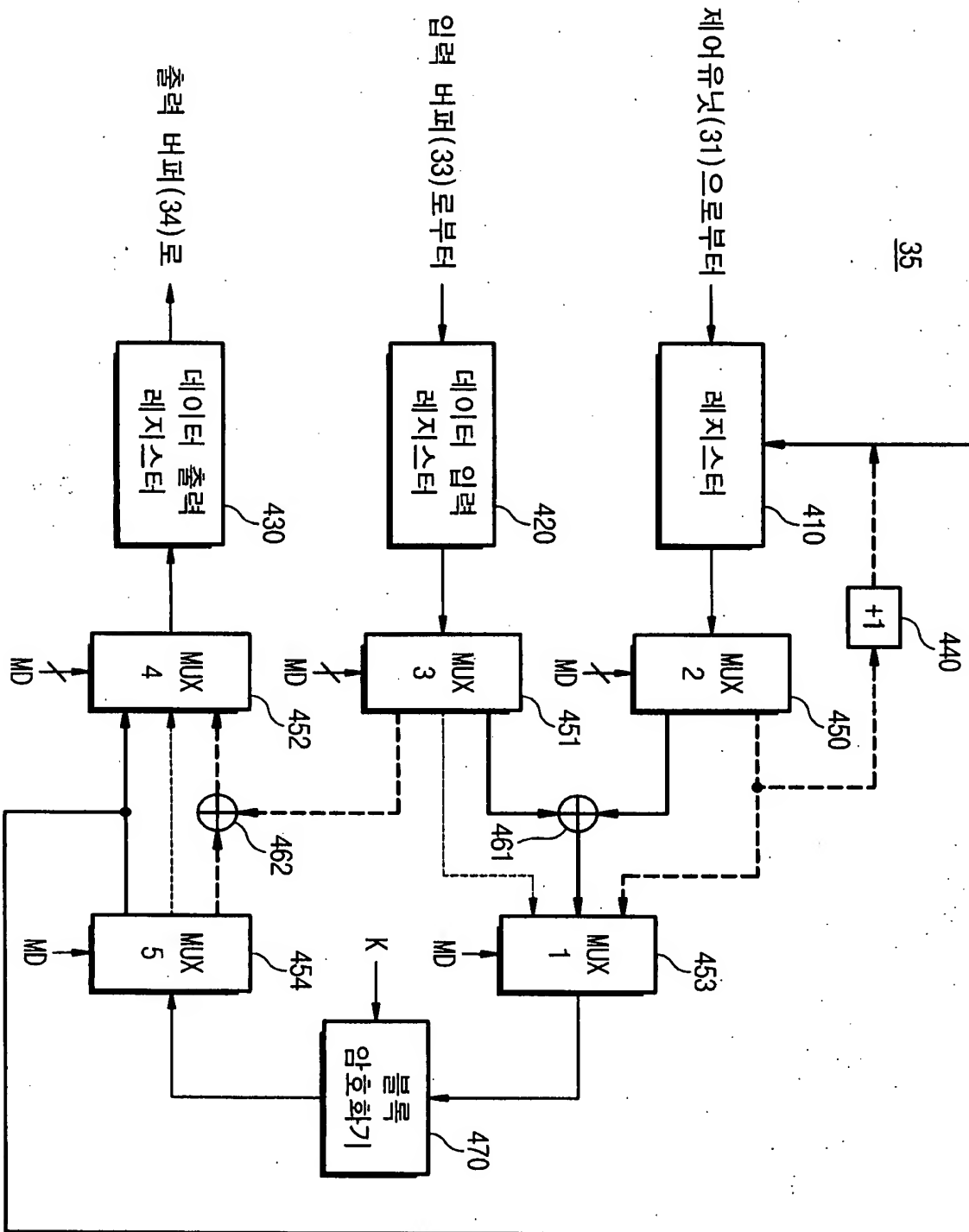
【도 4】

35b

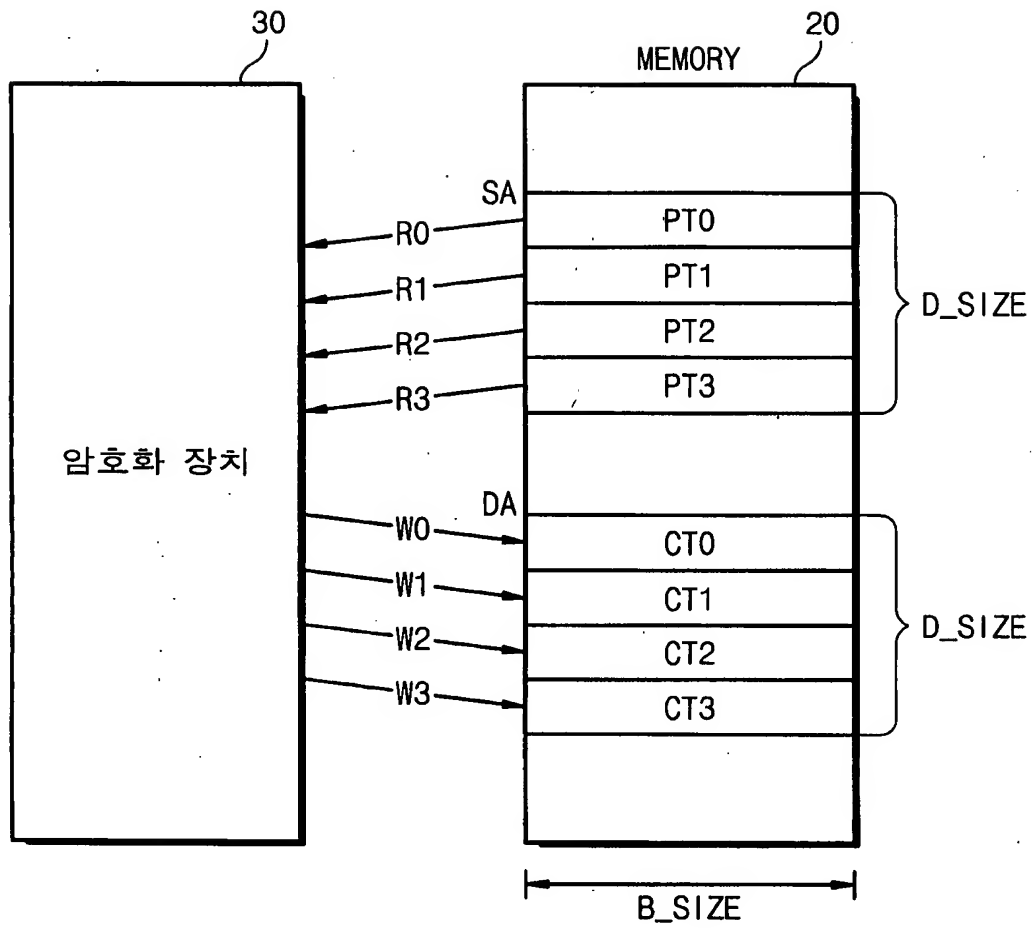
【도 5】

35c

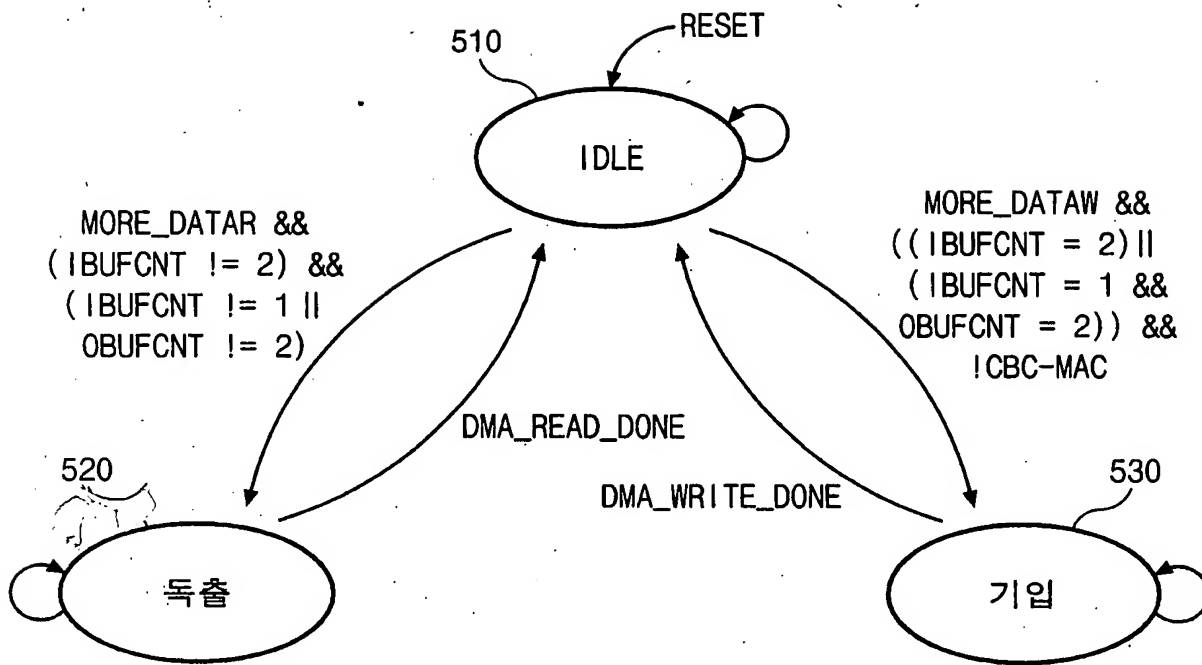
【도 6】



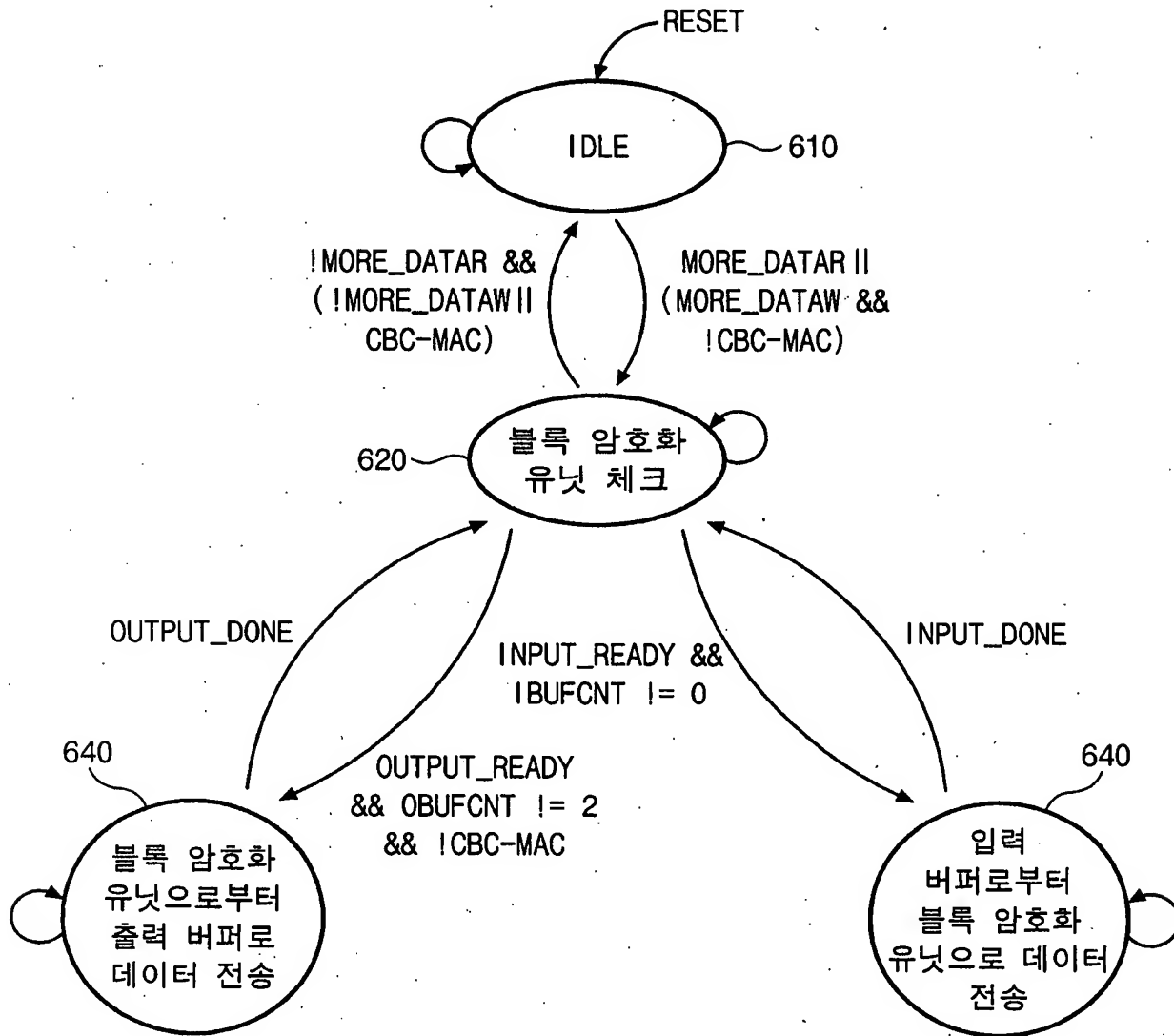
【도 7】



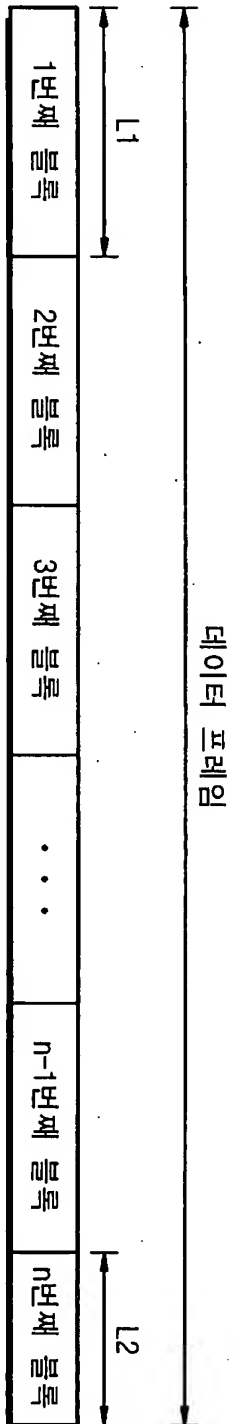
【도 8】



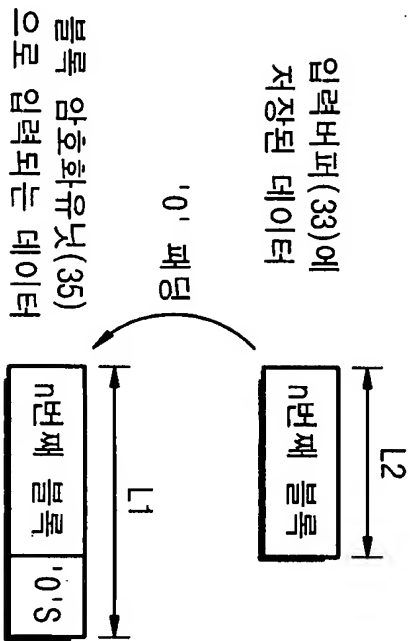
【도 9】



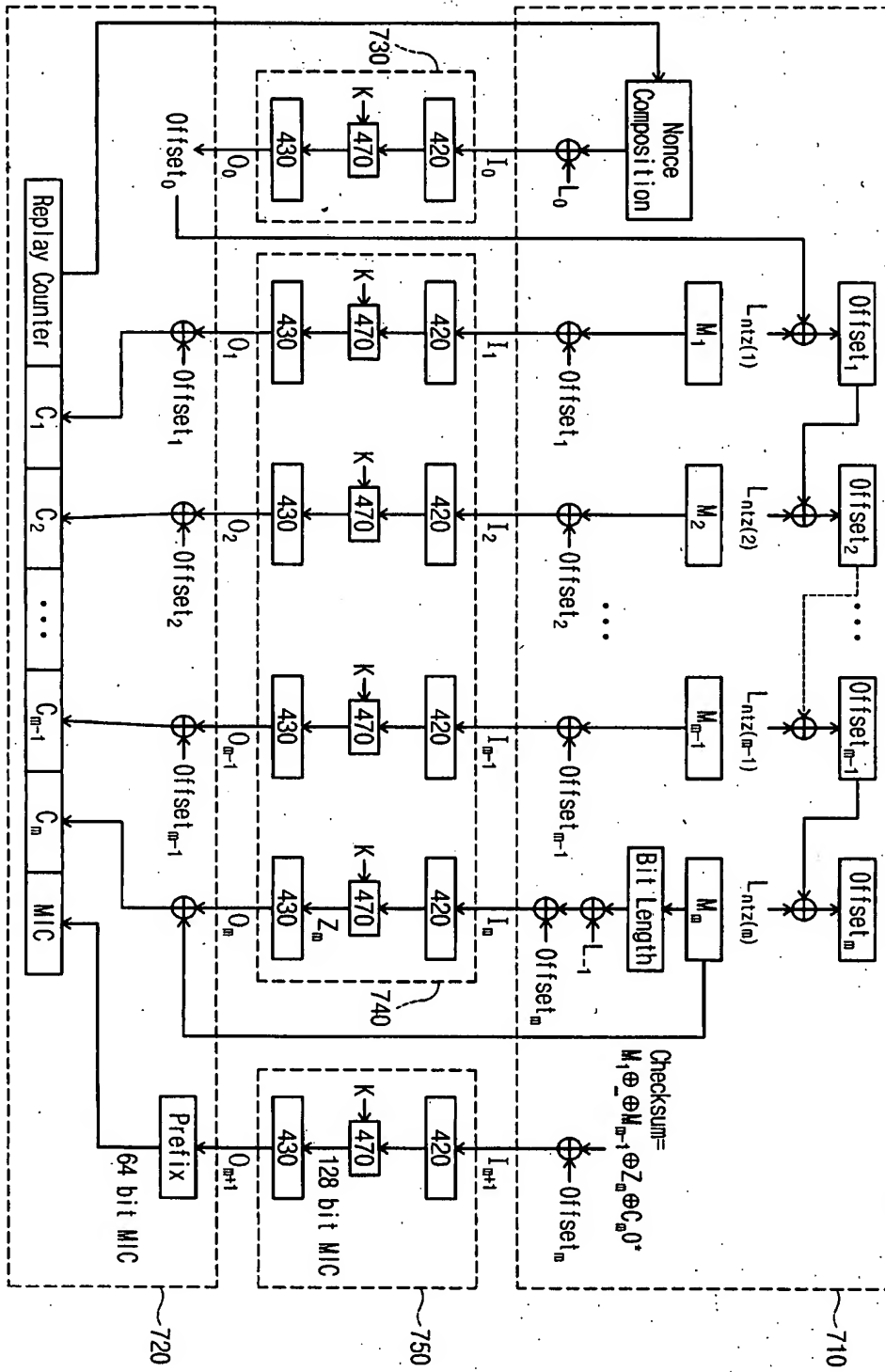
【도 10a】



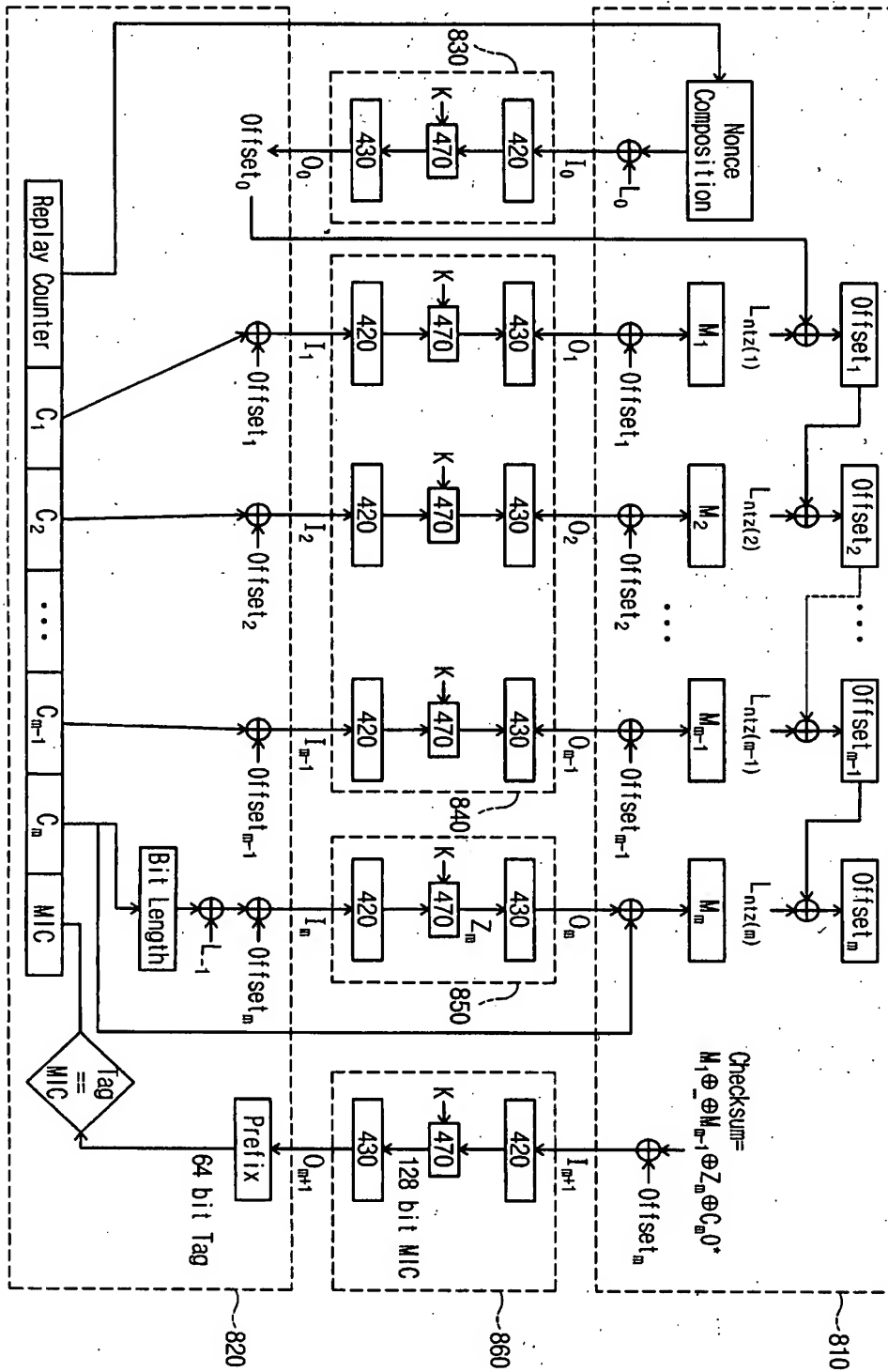
【도 10b】



【도 11】



【도 12】



【도 13】

